

5           **SYSTEMS AND METHODS FOR CONTEXT PERSONALIZED WEB  
BROWSING BASED ON A BROWSER COMPANION AGENT AND  
ASSOCIATED SERVICES**

10           **Inventor: Rajesh Bhatia, Nick Ganju, Anil Kamath, Prashant Kulkarni,  
Phoumrin Tan**

**Assigned to: BizRate.com**

15                       **BACKGROUND OF THE INVENTION**

              1. Field of the Invention

              This invention claims the benefit of co-pending US Provisional Application  
20       No. 60/227,558, entitled "Browser Companion Architecture", filed August 23,  
2000, the entire disclosure of which is hereby incorporated by reference as if set  
forth in its entirety for all purposes. This invention relates to accessing data over  
computer networks, and more particularly, to accessing information on the  
Internet, or World Wide Web (Web) with a personalized browser system. In one  
25       embodiment, the information relates to enhancing ecommerce functionality.

              2. Background Art

              Access to information on the Web is typically accomplished by invoking a  
program known as a Web browser. The browser acts as an interface between  
the user and the Web, accepting input from the user and obtaining data and  
30       services by sending requests to servers or other computers at other locations on  
the Web, and retrieving and displaying documents or pages from the Web. Fig. 1  
illustrates this scheme. Primary examples of commercially available browsers  
include Microsoft Internet Explorer and Netscape Communicator. Many  
traditional services and business transactions are now carried out over the Web

5 in this manner, and ecommerce has become a critical activity for businesses and consumers.

Although filled with valuable information and services, the Web is often difficult and time consuming to use with current browsing technology. In many ways, accessing the Web resembles the outdated mainframe network model with  
10 the browser playing the role of dumb terminal: much of the information needed is locked up in centralized databases, served up one page at a time to individual users who must know and remember where the information is that they are looking for.

To improve Web usage, search engines, such as Google™ and  
15 Altavista™ have been created to provide a more effective means for users to locate information on the Web. Browsers also include functionality to assist a user in relocating information by storing URLs for user-selected Web pages. This simplifies the user's future access to these bookmarked Web pages.

However, these improvements provide helpful features but are  
20 inadequate. Search engines are not personalized to a user and require a user to enter effective queries for accurate results. Moreover, searching for information through a browser requires a user to go to a separate search page to enter each query. A user must return to this search page to run subsequent queries. Moreover, standalone Web sites create islands of functionality and data that are  
25 not transportable across the Web. A user must manually navigate between sites and interfaces, which often involves repetitively logging in or form filling at each site and entering the same information repeatedly, such as inputting personal contact data, credit card data, etc. Attempts have been made to provide automatic form filling. These are reflected in Kikinis, US Pat. No. 5,794,259  
30 entitled *Apparatus And Methods To Enhance Web Browsing On The Internet*; Gupta et al., US Pat. No. 6,199,079 entitled *Method and System for*

5     *Automatically Filling Forms In An Integrated Network Based Transaction;*  
      *Environment*; Light et al., US Pat. Nos. 6,192,380 entitled *Automatic Web Based*  
      *Form Fill-In*; Kelley et al., US Pat. No. 6,247,029 entitled *Web Browser Form*  
      *Enhancements*; and Larsen et al., US Pat. No. 6,088,700 entitled *Automated*  
      *Forms Completion For Global Information Network Applications*. The foregoing  
10    patents are incorporated herein in their entireties for all purposes.

      Accordingly the state of the art is such that there still is inadequate real  
      interactivity between a user and the Web sites or other network sites accessed  
      through a browser. Users must directly act on information, rather than setting  
      intelligent preferences that then automatically act on the user's behalf. Users are  
15    increasingly demanding such advanced functionality integrated with other  
      functionality to automate and streamline a user's web experience. The other  
      functionality may include online search systems that provide more personalized  
      or context relevant results; online shopping information systems that provide  
      users with personalized or context relevant information; systems for storage of  
20    user profiles and preferences; systems for tracking, storing, and retrieving of  
      records and information related to users' online transactions; and systems for  
      automated data entry into online forms and data entry fields.

### SUMMARY OF THE INVENTION

25       The present invention addresses the foregoing needs by providing a new  
      paradigm of *context personalized browsing*. In this paradigm, in addition to the  
      conventional browser (e.g. Internet Explorer) and the Web server (e.g., that of a  
      merchant such as www.amazon.com), there is a novel software agent, referred  
      to herein as a "Browser Companion Agent or "BCA" for short. The BCA is  
30    associated with a conventional, main browser on a client system (which may be  
      referred to herein as the "main browser". Among other things, the BCA assists a

5 user by hosting and orchestrating one or more services that are relevant to  
information accessed through the main browser. The BCA assists users in a  
number of ways, for example, to shop faster and smarter by providing "what to  
buy", "where to buy" and "how to buy" services. It also displays messages based  
on user navigation and states merchant related information. It also may assist in  
10 automated online form filling. It also may be used to maintain records of a user's  
online transaction. The BCA is designed with a novel modular architecture so  
that the addition or deletion of services available through the BCA is readily  
facilitated.

15 BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 shows the prior art scheme for web browsing.

Fig. 2 is a schematic illustration of a general system of networked  
20 computers for implementing embodiments or aspects of the present invention.

Fig. 3 is a schematic illustration of general features of a computer system  
that may be used in implementing embodiments or aspects of the present invention.

Fig. 4 is a block schematic showing a general overview of system for  
context personalized browsing, in accordance with the present invention.

25 Fig. 5 shows an example user interface associated with a main browser, in  
accordance with the present invention.

Fig. 6 is a block schematic showing a general overview of the architecture  
of a browser companion agent for use in a system for context personalized  
browsing, in accordance with the present invention.

30 Fig. 7 is a diagram chart that illustrates objects and related features,  
according to an embodiment of the present invention.

5           Fig. 8 is a diagram that illustrates objects and related features, according to an embodiment of the present invention.

          Fig. 9 is a diagram chart that illustrates objects and related features, according to an embodiment of the present invention.

          Fig. 10 is a diagram that illustrates certain features of an automated form-  
10   filling service, according to an embodiment of the present invention.

          Figs. 11a-b is a flow chart that illustrates certain features of an automated form-filling service, according to an embodiment of the present invention.

          Figs. 12 is a flow chart that illustrates certain features of an automated form-filling service, according to an embodiment of the present invention.

15           Figs. 13 is a flow chart that illustrates certain features of an automated form-filling service, according to an embodiment of the present invention.

## **DETAILED DESCRIPTION OF THE INVENTION**

### **20   General Overview of Features**

#### ***1. Browser Companion Agent:***

          The agent that orchestrates contest personalized browsing on a host system. The agent sits inside or is otherwise coupled to a "content browser",  
25   which could be a standard web browser, a browser for an e-book system, etc. The content browser provides access to content, and the BCA monitors the content which is provided by the browser. The BCA resides on the host or client machine, which most often would be, a computer being used to "surf" the web. This agent might take the form of a COM object executing within Internet  
30   Explorer.

5           2. *Tracker object*

This object is used by the BCA to monitor the content in the browser. Requirements for this object include notification of changes in browser content. Effectively this object listens to the browser for changes in content and delegates processing to other objects depending on what the content is. This object resides on the host machine. This object may be implemented inside of the BCA object as another COM object.

          3. *Proxy Object*

15           This object is used by the BCA to modify content sent back to the browser component. If the Tracker object determines that certain content requires a certain type of processing, the proxy object manages transfer of other content into the BCA object for assistance with this processing. This object resides on the host machine. This object may be implemented as a COM object which provides network services to other COM objects.

20

          4. *Type-and-learn (TAL) Object (For use in connection with an optional automated form filling service module)*

          This object resides on the remote machine, unlike most of the other objects. It maintains a registry of all the communication made by the BCA on a host machine. Depending on the interactions and content provided to this object, the TAL object modifies its internal state, to improve services provided by the RSC (below). This object could take the form of a database with user profiles correlated to websites visited by those users.

30           5. *Remote Services Code (RSC) object (Implemented under Application Service Providers, discussed below)*

GANZLAW, PC  
PO Box 10105  
Portland, Oregon 97296  
Phone: (503) 224-2713  
Docket No.: BIZ/01-0008  
Express Mail Label No.: EL627039647US Deposited: August 20, 2001

5           The RSC object is a server-side code which acts, based on the user, what content is being viewed, state parameters, and possibly other factors, sends back the appropriate correlating code and/or data to the BCA. The code often will correlate to a specific service module for the BCA

10           6. *Code Assistance Objects*

          These are pieces of executable code that are sent back by the RSC in order to assist with browsing sessions. The code often will correlate to a specific service module for the BCA. They are customized depending on the state of the service, and customized depending on the user accessing the objects. These objects will often take the form of JavaScript code. These objects are produced on the RSC and sent over to be received and executed in the BCA. An example implementation of these objects would be JavaScript source files as text.

20           Detailed Discussion of Features

          The systems, methods, and features of the present invention may be implemented through a networked computer system, such as a personal computer on the Internet. As person skilled in the art will appreciate from the teachings herein that the novel Browser Companion Agent ("BCA") of the present invention may be implemented through a variety of software protocols, platforms, or environments. For the sake of illustrating its salient features, the present invention is generally described in terms of implementation of C++ programming language in connection with an Internet Explorer browser. However, this is intended to be a non-limiting example, the appended claims defining the scope of the invention.

5           The present invention is intended for use with a content browser such as a web browser or e-book browser. Also for non-limiting purposes of illustration, the present invention is generally discussed in terms of a conventional web browsers, such as Microsoft Internet Explorer or Netscape Communicator, using known standards and protocols applicable to a browser system. For example, the standards and protocols for implementing the present invention in an Internet Explorer or Netscape Navigator are known to persons skilled in the art and are freely available from providers of browser systems, for example.

- MSDN Online Web Workshop: "Creating Custom Explorer Bars, Tool Bands, and Desk Bands"; MSDN "Browser Helper Objects: The Browser the Way You Want It", available through Microsoft's online technical library at <http://msdn.microsoft.com/library>; and
- Netscape's DDE (Dynamic Data Exchange) Implementations: <http://developer.netscape.com/docs/manuals/communicator/DDE/abtdde.htm>

20           The BCA uses a novel architecture that can display information related to a browser's content. The BCA monitors content pages being viewed by a user in a main browser window and determines the content or subject and can display related information about the page, such as news, products, specials, discounts, and comparison information and pricing. As a matter of architecture, the agent may be a thin client composed of objects. In one possible embodiment, the BCA has certain common components that may be shared by multiple service modules to enable services (e.g., purchase tracking, form filling, etc.) Each service may rely on the common components to execute their specific functionality. In one possible embodiment of BCA architecture, the BCA is modularly composed of Component Object Model (COM) objects. New objects representing services can be added incrementally or existing services can be enhanced to offer additional functionality through the novel architecture of the BCA.



5           The BCA can also store personal information of a user and automatically fill this information into pages that request it. It may also store other information related to a user's online activities. For example, it can be used to store information related to an online transaction, such as an order confirmation.

10           Due to the relatively limited resources of most client systems, the BCA works in conjunction with a central server to deliver services to users of the client systems with BCAs. For example, the BCA, through a Tracker Object (discussed in detail below), receives related data or information from a central server after the BCA sends data about a specific page on the client system's main browser. The server uses this data from the page to determine the subject or context of  
15   the page. Once this is determined, the server may send related information to the BCA from its database or other resources. For example, if a user is looking at a purchase form of an online merchant, for example, Amazon.com, the BCA sends relevant data (e.g., domain, user identity) to the server, which then returns the domain specific information such as name-value pairs, form filling scripts,  
20   personal data of a user, so the BCA can automatically fill in the form. For efficiency, a user's personal data is retrieved only once from the server during a session. As another example, the page could contain information on products the user may be interested in purchasing. The page could be parsed by the server or the client for the product data, the data may be used by the server to  
25   return comparison product information to the user.

          The server functions noted above may be associated with or included as part of an infomediary Web site. Generally, an "infomediary" provides information about products and merchants; maintains information about user shopping profiles or preferences; provides buyers access to online merchants;  
30   and/or otherwise facilitates transactions between online buyers and sellers. One such infomediary is BizRate.com ([www.bizrate.com](http://www.bizrate.com)), the assignee of the present

5 invention. The server functions of the present invention could also be implemented through other online service providers such as Web portals, or any other Web site operator. Bizrate.com may be used herein as the example Web site from which Internet users could download the software for the BCA and who would provide data and applications used in the systems of the present invention.

10 The methods of the present invention may be stored in or implemented through computer readable media associated with a single computer system or a distributed computer system. Looking more particularly at the systems and methods for implementing the present invention, Fig. 2 shows an overview of a system 1 of a plurality of networked computers, according to the present

15 invention. System 1 includes a Central Computer System ("CCS") or central server 2, one or more consumer or client computer systems 3, a plurality of content provider computer systems 4, such as may be controlled by online merchants. The CCS computer system would typically be an infomediary's computer system and would execute one or more features of the present  
20 invention. For example, as discussed below, it may include a server with downloadable BCAs; it may update BCAs installed on client computers; it may serve data or code to a client computer in connection with the operation of a BCA on the client computer; it may provide new service modules for installation in a BCA on a client computer; it may store personal data or transactional data for  
25 user's of client computer systems with BCAs; it may provide ecommerce and shopping assistance services to users of client computer systems; and/or it may provide security and authentication services to client computer systems.

Computer systems in system 1 may be networked over LANs, WANs, VPNS, the Internet or other known network systems using known protocols for such

30 systems, including TCP/IP.

5            Fig. 2 illustrates features that may be typically found in a computer system  
2, 3, or 4 in system 1. As used herein a "computer system" generally means a  
set of hardware and software components as is known in the art. A computer  
system 2, 3, or 4 generally includes: central processing unit ("CPU") 2.01;  
memory (ROM and RAM) 2.02, e.g. a hard disk drive and associated RAM  
10 devices; programs 2.03 in memory, including operating system 2.31, web  
browser applications 2.32 and/or web server applications 2.33, user applications  
(modules or agents) 2.34, database management system (DBMS) 2.35,  
communications and network applications 2.36, e.g., email, and network  
interface systems. Memory 2.02 would also include stored data 2.04.

15            A computer system may include one or more of the following input/output  
devices: keyboard 2.7, mouse 2.8, display monitor 2.9, read/write drive 2.10 for  
removable media, e.g., DVD drive, PCMCIA card 2.11, modem 2.12, backup  
storage device 2.13, e.g., tape drive, audio capture device 2.14, e.g.,  
microphone, audio output device 2.15, e.g., stereo speakers, still or motion  
20 image capture device, e.g., camera 2.16, and an associated computer device for  
data upload or download, e.g., PDA 2.17, and/or a printer 2.19.

            It will be understood by persons skilled in the art that a computer system  
2, 3, or 4 may also be in the form of a Personal Digital Assistant (PDA), a  
wireless computer system or device capable of communications over the Internet  
25 or other network, or a computer terminal (thin client) or Internet appliance  
capable of such network communications.

            A computer system 2, 3 or 4 in system 1 may be in the form of a  
distributed system not, under a single housing or roof, but having the distributed  
components logically coupled and under the operational control of a party.

30            Computer systems of system 1 may execute a conventional operating  
system 2.31, including, but not limited to, Microsoft Corporation's Windows (sm)

5 3.x, 95, 98, 2000, ME, NT or DOS operating systems; Linux operating systems;  
Sun Microsystems' Unix Solaris operating systems; IBM's AIX Solaris operating  
systems; SGI's IRIX Solaris operating systems, Apple OS, BE OS or other  
operating systems, including, but not limited to Window's CE, Palm OS, etc. for  
10 PDAs, cellular devices, web devices, or other computing devices. The possible  
software components of a computer system 2, 3 or 4 described herein, including  
operating system and application software, may reside on one or more hardware  
components providing addressable memory for the computer system.

15 Data and information may be formatted by a party's computer system for  
printing, viewing, transfer to another computer system, storage in any of many  
known formats, including, Adobe PDF; rich text (RTF); SGML formats including  
HTML, DHTML, XHTML, and XML; ALLAIRE Cold Fusion CFML; MICROSOFT  
ASP; JAVA Script MICROSOFT Word, Excel, Access; bitmap, JPEG, GIF, and  
other graphics file formats. It could also be presented in known audio or video  
formats such as WAV, MPEG, and MP3 formats.

20 The electronic data file may be served via known protocols, such as  
TCP/IP, and formatted in known formats of standardized mark up languages  
such as HTML and XML. Web documents generally means electronic  
documents that may be presented through conventional web browser  
applications such as the INTERNET EXPLORER series of web browser  
25 applications by Microsoft Corporation, or the NETSCAPE COMMUNICATOR  
series of web browsers by Netscape Communications Corporation. As used  
herein, "client" means a computer system that makes a request to or receives  
fulfillment of a request from another computer system. "Server" means a  
computer system that receives and/or fulfills a request from another computer  
30 system. Also, as used herein the term "computer system" may be used

5 interchangeably with "computer", "computing device", "machine", "system having computational resources" or like terms.

A computer system 4 executing features of the present invention may include various external interface applications for converting incoming data, such as form data, to a format for communication with the respective application  
10 module or database. The system may also include various external service modules for obtaining information in a format required by the external service. The system may also include a notification module. This module handles notification and electronic mail communication with client systems. The system may include a database interface module. This module handles all interface  
15 requirements between application modules and the databases. For example, the database interface application manages (a) the simultaneous connections to the database such that the total number is maintained within applicable license agreements; and (b) any integrity problems in interactions between application modules and the databases. Thus, this module allows the system to be  
20 connected to different types of databases such as relational databases or object-oriented databases. Another function that may be included in the system is a consumer profile/settings module. This module receives the consumer profile and settings data sent to the system and performs intake operations on the data so that it may be used in other modules or assigned to databases.

25 The present invention contemplates that different parties may create and manage database records, input and manage data, upload files, communicate, compute data, collaborate, and exchange things with each other through their respective computer systems. The data and files may be transferred or presented to computer systems by participating parties via user interfaces  
30 developed in Standardized General Markup Languages (SGML), scripting languages, and other known languages or protocols. Such languages and

5 protocols include MICROSOFT ASP, CFML, JAVASCRIPT, JAVA, COM, CGI,  
HTML, DHTML, XHTML and/or XML. To provide the foregoing functionality, the  
user interfaces would typically be in the nature of electronic documents such as  
web documents or email documents that, through appropriate computer code  
from such languages, are associated with databases, web servers, web  
10 browsers, file servers, network communications links, email programs, etc. The  
electronic documents could be static documents stored on a server or database  
or documents created dynamically through server function calls, for example.

Databases may be based on any number of known DBMSs, including  
hierarchical databases, network databases, relational databases, and object-  
15 oriented databases. Suitable DBMSs are widely available and include Oracle,  
Sybase, Microsoft SQL Server, open source MySQL, and DB2. One suitable  
database system is a relational database based on Structured Query Language  
(SQL). A suitable SQL database would typically include a Data Definition  
Language (DDL) for creation of a new database and new objects within an  
20 existing database; a Data Manipulation Language (DML) for processing existing  
database objects; and a Data Control Language (DCL) for administering  
permissions and other security-related issues. The data processing is handled by  
a database engine and can be accessed by various querying means, including  
Boolean logic evaluation, proximity calculations, and fuzzy logic evaluation. The  
25 search engines and databases of the present invention may also be custom  
designed, the design process and features being well within the skill of persons  
in the art. U.S. Patent No. 5,983,220 of Schmitt, entitled "Database Evaluation  
System Supporting Intuitive Decision in Complex Multi-Attributive Domains Using  
Fuzzy, Hierarchical Expert Models" discloses, among other things, systems and  
30 methods for constructing and querying databases and returning and presenting

5 results of database queries. The '220 patent is hereby incorporated by reference for all that is disclosed therewithin as if set forth herein in its entirety.

The computer systems suitable for use may include or otherwise be directly associated with a web server and associated database. As used herein, a web server generally refers to a computer system that hosts software that  
10 enables access by remote computer systems or terminals via the Internet or other network transport. The web server may permit one or more of the following: presentation of data and information, typically in the form of web documents or streaming media; data and information storage and transfer; and other transactions or processes that are capable through web servers. Suitable  
15 web server software is widely available and includes Apache Software Foundation's Apache Server web server; Netscape's FASTTRACK series web server, Microsoft Corporation's Internet Information Server, IBM's WebSphere, and Sun Microsystem's JAVA WEB SERVER. A web server in CCS 2 may run appropriate scripts to capture data, to process captured data, or to present  
20 processed data. A web server of CCS 2 may receive into or present data from an associated database or other data source to remote computer systems 3, 4, or other remote system, over a network 5. The data may be presented in any known format or mechanism, including HTML, XML, PDF, CFML, MICROSOFT ASP, and as graphic, audio, video and multimedia formats, and others.

25 The foregoing embodiments and features are for illustrative purposes and are not intended to be limiting persons skilled in the art capable of appreciating other embodiments from the scope and spirit of the foregoing teachings. As a general note, in the text and figures, the portion of referenced components and objects given as "...eB..." or "...eBoodle..." is an arbitrary designation used to  
30 designate a component or object ("eBoodle") described herein. For example, "CeBObject" is a Browser Helper Object (BHO), as noted below. The present

5 invention provides novel applications for BHO technology, although BHO  
technology is generally known in the art. See, for example, Semenzato, US Pat.  
No. 5,903,728 entitled *Plug-In Control Including An Independent Plug-In Process*;  
Meyer, US Pat. No. 6,266,058 entitled *Apparatus And Method For Linking*  
10 *Browser Bars With Active Documents For A Browser*, the entire disclosures of  
which are hereby incorporated by reference as if set forth in their entireties for all  
purposes.

Fig. 1 shows the conventional (prior art) model for web browsing. A  
browser 10 of a client computer system 3 could communicate with web servers  
4a, 4b, and 4c of different parties. In this example the client computer system is  
15 operated by a consumer and the web servers are operated by online merchants,  
at domains such as amazon.com 4a, buy.com 4b or barnesandnoble.com 4c.

Fig. 4 describes a novel embodiment of a system for context personalized  
browsing according to the present invention. In the embodiment shown, client 3  
includes a BCA 12 associated with a browser 10. The BCA communicates with  
20 the browser. It may also communicate with a remote computer systems such as  
a CCS 2. The BCA monitors the pages being seen by a user on the browser  
window to determine the browsing context. The BCA then sends information  
about the browsed page to the server. The server uses this information about the  
browsed page to determine the context. The server can avail itself of data stored  
25 in the database. It may also go to other resources, such as the Web to seek  
relevant information needed in providing assistance. The server uses these  
resources to provide the BCA with pages, scripts and data that the BCA uses in  
services to enhance the browsing experience. For example, if a user is looking  
at a form within the domain of amazon.com, the BCA sends the information  
30 (domain, user identity) to the server, which then returns the site specific name-



5 value pairs, form filling scripts, personal data of a user, etc. so that the BCA's automated form filling service can fill the form.

10 In a preferred embodiment, CCS 2 includes certain applications that may communicate with the BCA 12 and/or browser 10. For example, as indicated in Fig. 4, the CCS may include a comparison shopping search application 14; a "wallet" application 16 that maintains personal data of users of client 3 and enables use of the data in online transactions with third party sites such as 4a, 4b, and 4c; and/or a transaction tracking application 18 that acquires and maintains records and information related to online transactions by the user of a client system. These are example applications or services. It should be  
15 appreciated that there may be service additions or deletions depending on the preferences of a system administrator or designer. Databases may be associated with the applications, as needed. For example, database 22 could be used to store data used by comparison shopping search engine 14. The data categories could include products by category, product features, product prices, product availability, etc. Database 24 could be associated with the wallet  
20 application 18 and/or transaction tracking application 18. The data categories used by these applications may include user name, address, credit card number, merchant name, merchant receipt, etc.

Looking at Fig. 5, the interaction between a user and the services  
25 provided by or through the BCA may be coordinated through a user interface ("UI") 26. In this example, the BCA UI is a thin docked window or bar 26 associated with a main browser window 28. The BCA UI preferably provides a display area 29 adjacent to the browser pane. It may be a child window within the browser window, and it can be used to display information and interact with the  
30 user in much the same way as a main browser. The BCA UI may occupy any desired area of a computer display. The bar is not limited to the size or

5 positioning of that shown. Generally, the BCA UI would occupy a small area of  
the display so that it is subordinate to a main browser. Further, more than one  
BCA UI may appear at a time for different services. Using standard programming  
techniques, the BCA UI may be configured to change proportionally with a user's  
changes to the size of a main browser window 28. It may also be configured to  
10 be moveable over a display screen.

The BCA UI may display HTML, other SGML based pages, and other  
browser displayable pages related to BCA services and also provide input areas  
and controls that a user can invoke to get specialized assistance. For example,  
in a transaction-tracking service, the BCA window 29 could display the cash-back  
15 earned when a user buys an item through an infomediary's website. As another  
example, in form-filling service mode, a user could use the input areas 31, 33 of  
the window to change the credit-card that is used to buy the item. In connection  
with a shopping service, the bar advantageously may provide an interface for a  
shopping service that allows a user to quickly and easily search from any website  
20 location on the main browser, without returning to a search home page to begin  
another search.

Figs. 6-7 show one possible embodiment of a scheme 30 of objects and  
data flow associated with service modules 56, 58, 60 for a BCA 12. The BCA  
may be adapted to start on initiation of a browser, for example, Internet Explorer.  
25 A Browser Helper Object (BHO) is initiated in a session by a remote computer,  
such as the CCS. Upon initiation, the BHO reads a client-side index of  
functionality that may be called upon by the BCA. In the case of a Microsoft  
Windows environment, the BHO would read the client system's registry. For  
example, in Fig. 8, BCA objects, functions, calls, and events are generally  
30 illustrated. The class name for the BHO 32 is CeBObject. On every new instance,  
the browser calls 34 CeBObject::SetSite method and passes a handle to the

5 main browser. The BHO reads the client system's registry to determine the need to auto launch 36 the BCA. This registry setting is a persistent user preference of launch state and the companion is launched if the auto launch state is enabled to be on.

On the launch of a BCA, CeBoodleCommandBand::Setsite method is  
10 called by Internet Explorer. The parameter passed to this method is an indirect interface to the browser object. The class CeBoodleCommandBand 38 stores this interface in a global interface holder and creates an instance of View class control. The View class control is CeBoodleView 40 which embeds a BHO. The View class creates the browser objects and shows the index page. This index  
15 page calls the initialization method from within the HTML. The initialization method creates the Tracker object and calls the Start method.

The Tracker object tracks and monitors the browser window (which may also be referred to as the "Host") to determine an identifier for the location of content on the browser. For example, in the case of a web browser, the identifier  
20 would be the domain, e.g., URL, of content on an associated main browser window. TheTracker object has the logic to detect a change of page in the window. The Host includes facilities to notify or provide queried access to change content in the host display. For example, if a user navigates from a current web page to another, the Tracker detects the change between pages. Fig. 6 illustrates  
25 an example of a class object CeDomainTracker 42 that maintains a collection of BCA services. The Tracker object is created by the View object and is started by calling the Start method. This method does the following:

- In this example, the Tracker object uses a "listener interface" to receive automatic notification of page changes from the Host. For example, it  
30 connects to the main browser object to receive events from the browser, using the IconnectionPoint interface.

- 5 • Verifies in the Host environment whether there are services to be performed. For example, the Tracker object looks at a list of services on a remote machine (e.g., CCS2) and compares the list to a locally maintained list to determine what services or other functions to perform. In the Microsoft environment, there is a scan of registered services, examining the machine
- 10 registry under HKEY\_LOCAL\_MACHINE\SOFTWARE\EBOODLE\Bsvcs and loop for all the class identifier (CLSID) found and a check if the CLSID belongs to CATID\_EboodleService and passes an authentication check. On matching of the criteria above, the service is created and inserted in the list. There are multiple key and value pairs for registry
- 15 HKEY\_LOCAL\_MACHINE\SOFTWARE\EBOODLE\Bsvcs, and are filled by a script. For example, the script may be JavaScript in an HTML file that is executed once for the entire session in the View class.

The MainBrowserProxy object is used to set the registry. The keys may be a sequence of numbers beginning from 1 and values are CLSIDs for every

20 number. The key value pair represent the sequence in which the services should be created and called by the Tracker. The CATID\_EboodleService is an interface defined as a standard and which may be implemented by all services. In this way, the Tracker may communicate with all services using a common set of Application Program Interfaces (APIs).

25 The Tracker waits for the browser to call back on an event. The events may be predefined. For example, in the case of BHOs, Microsoft has various defined events, and the Tracker adheres to the specified rules and guidelines for such events. The Microsoft BHO events which the Tracker may respond to include one or more of the following:

- 5     • BeforeNavigate2 (User has expressed interest in content, but the user's machine has not communicated the interest to another machine with the content page)
- Navigatecomplete2 (After communication has started, but not necessarily before the communication is complete)
- 10    • Downloadcomplete (Once download is complete, but before display to user)
- Documentcomplete2 (Download complete and page displayed)

15       In the example of Fig. 6, the modularity of the BCA is shown. BCA 12 includes service collection 49 and data object collection 50. Data object collection 50 includes a collection 52 of one or more data component objects 53 consisting of data objects and/or a collection 54 of one or more script component objects 55. Data objects 53 may be provided to hold data for use in services 56, 58, 60 associated with a BCA. The Script Objects 55 may be provided to store a collection of scripts that are sent by a server and used by the BCA in providing specific services. Since these scripts are dynamically generated by the server and made available to the agent service providers, this is an advantageous way of adding new or modifying existing services, without a fresh install of the agent. New data structures with corresponding scripts can be added to the agent to reflect changes to websites that users visit, to provide enhanced services, or to otherwise update data for use by a service. Advantageously, since user specific data will typically be server resident, it is not machine dependent.

25       For example, regarding a form filling service (described below), the data could relate to details of a Web page on a main browser. For example, each Web page can contain frames or no frames and each page or frame can contain one or more forms. Each form can consist of one or more elements. Frame details, information about forms and form properties contained on the page or in a frame can be contained in a data object. For convenience, unless context

5 indicates otherwise, the term "page" shall mean page, frame, or other discrete  
browser displayable element. The script object components 55 may be used to  
instruct the client system relative to data in a data object or to implement service  
features that can use or interface with the scripts. In the case of both the script  
objects and data objects, common interfaces may be provided so that multiple  
10 service modules may access any given data or script object in Data Object  
Collection 50.

Looking at Fig. 8, to store a generic page structure in the memory, the  
following data objects may be required:

- 15 • **Form Data Object:** CeBFormDataObject contains a map of key value pairs  
which represents the form element name and value.
- **Frame Data Object:** CeBFrameDataObject contains a collection of form data  
objects
- **Page Data Object:** CeBPageDataObject contains a collection of frame data  
objects.
- 20 • **Collection of Page Data:** CeBEnumPageDataObject contains a collection of  
page data objects.
- **Site Data Object:** Complete data for a site resides in this data object.  
CeBDomainDataObject contains a collection of page data and a map of key  
value pairs. A portion of the generic data about the site is stored directly in  
25 this data object since it does not belong to any page. For example, the  
assignee of this patent document maintains on its website systems various  
categories of information for online merchants. The information categories  
include, for instance, consumer ratings data for various online merchants,  
merchant group rating (e.g., gold, silver, etc.), merchant participation in rebate  
30 programs, etc. The data representing such information may be stored in  
CeBDomainDataObject.

- 5 • **Collection of site data Object:** CeBDomainDataObjectCollection object keeps a collection of data for 'n' sites , where 'n' can be maximum size set in the object. On addition of a new item, the Least Recently Used (LRU) technique is used to figure out which item needs to be associated with the new one. A logical site can belong to multiple domains. For example,
- 10 www.shades.com uses www.cartserver.com for a checkout process, and the data for the shades and cartserver Web sites is the same. The collection object contains two entries: one for shades as key and one for cartserver as key. The value object for these two keys is the same pointer and hence the same memory location. On addition of a new site data object, the total items in the list is compared with the threshold, which is set using registry entry and
- 15 can be modified as needed. If the total is greater than the threshold, then iteration through the list is performed to get the least recently used site. The list is again iterated to get all aliases for the site that is the least recently used and the site and aliases are then deleted from the list for a new site to be
- 20 entered.

Fig. 9 shows the interaction between different components of the BCA. The BCA may be comprised of C++, JavaScript and HTML code. The components are View, Tracker, and BHOs. All of these communicate with each other and have access to data on a need basis. For example, scripts such as

25 JavaScript, in the services have access to data stored in the memory, access to main browser object, etc. The main browser object is a core component of Internet Explorer and is needed by all the components. The BCA may access to the main browser object by using a BHO or a "band object. The object is stored in a global interface holder, one of the helper objects described below. The other

30 helper object is a Browser Proxy Object.

## 5 **Helper Object Components**

- **Browser Proxy component** : CebMainBrowserProxy acts as a proxy for the main browser. The objects expose APIs to get the main browser object, registry access, communicate between a web site and the companion. For example, when a user goes to a site to change his or her personal profile (e.g., to update addresses and credit cards), the message is sent to the BCA via the browser proxy component.
- **Global object holder component**: CeBGlobalInterfaceHolder is a singleton object, which can be used to make an object globally available. An object needs to be registered with the global object holder. The holder may be instantiated anywhere in the same process and is used to get access to registered objects.
- **The View component object**: The View object is the presentation component that embeds a BHO to render HTML pages. It also provides APIs to persist JavaScript variables in memory so that they are available from one page to the other. For example, state messages are saved in the memory so that when a user navigates from one tab to another, the messages are not lost. On every new page download in the View, if it exists, the stored message is displayed. The View object also provides a communication channel between a user action on HTML and services.

## **Server Request Gateway**

The Server Request Gateway 62 interfaces between service providers and the server. On one hand, it routes requests, data and pages from the service providers to the server. On the other, it interprets server input and stores the scripts received from the server in the script collection object, the data in the data



- 5 collection object. Related pages or information may be displayed in the BCA UI or the browser.

### **Server Architecture**

- 10 The CCS 2 or server side has an API for interfacing with the BCA 12. The server receives requests, data and pages from the agent and sends data, pages, and scripts back to the agent. In one possible embodiment, the server architecture has certain components that may be shared by a variety of application service providers for providing services. Examples of application service providers are provided below. New application service providers can be added that use the shared components to offer new services to a user.

### ***Database***

- 15 One component of the server architecture may be a database. The database generally stores at least two kinds of data -- profile information about a user and domain specific information about sites a user may visit through a browser on the client computer associated with a user. Domain data may be  
20 URLs or other addresses of pages viewable through a browser. User profile data helps in the personalization and the domain-specific data is used for contextual information that can be shared by all the application service providers.

### ***Search engine***

- 25 The server architecture may also include a search engine that can access websites and obtain data to populate databases or data objects associated with BCAs on client systems. Preferably, the search engine is a multi-threaded search engine that can go across the Web concurrently to multiple sites on demand and obtain desired pages from sites. The multi-threaded engine manages the streams of data that are received from the threads and passes

5 them on to, for example, databases, service modules associated with BCAs, or other data users.

### ***Data extraction engine***

The server architecture also may include a data extraction engine for extracting information from Web pages retrieved by the search engine, from a client machine, or other source. The data extraction engine uses parsing  
10 modules and parsing libraries to assist in the extraction of the relevant information.

### ***Parsing modules***

To extract data from a page, the page needs to be modeled to understand  
15 its underlying structure and the information that it represents. The parsing modules may be individually written for each domain and each page type based on the page model and the information that needs to be extracted. The data extraction engine uses these parsing modules to extract desired information. The modules and the data extraction engine may use a library of commonly used  
20 methods that are shared and serve as a repository of domain independent parsing functionality.

### ***Application Service Providers***

As noted above, the server architecture may include application service providers that enable the various service modules associated with BCAs on client  
25 systems. The architecture of the present invention advantageously provides the flexibility to allow additions and modifications of services over time. The application service providers can avail themselves of several important components like the data extraction engine, the search engine and/or the database in interpreting the data sent by the BCA and in providing the agent with

5 tools for context personalized browsing. The applications can also leverage the models that have been developed for the different page types at different stores.

### ***The Services***

10 The present invention contemplates various services, including automatic form filling; transaction tracking; comparison shopping, context-related-messaging, automated hyperlinking and information retrieval (see US Pat App. No. 09/881,353, which is incorporated by reference below); presentation to consumers of online offers, the offers being determined by the content on a user's browser or the users interaction with the content; searching service that  
15 directs information to a user dependent on the users browser content or interaction with the content; searching service that monitors information keyed by the user into a search interface; chatting service that allows users to chat with each other, for example users with a common page on their browsers could be enabled to chat with each other; buying power service that monitors for common  
20 interest in content page about a product or service and provides discounts or other offers to users based on a collective level of interest. The modular architecture of the present invention provides a convenient and efficient architecture for including any one or more the foregoing services or any other service that can be dependent on the context of content on a users browser. In  
25 addition to implementing services individually, services may be implemented jointly. For example, Form filling and Transaction Tracking service may be implemented as one service. This service creates three internal connections to the server using the BHO. The connections hold the data, JavaScript logic for a single site, and user.

30 The services provide specialized functions to a user in connection with a user's browsing activities. The Tracker and the BCA UI interface with the services

5 through the BCA services interface. The service providers may use a Server  
Request Gateway at the other end to communicate with the server. They also  
have access to the collection of Script and Data Objects. Each service module  
may be a COM object that a client computer may download from the server and  
plugged in to the BCA where they are available until they are uninstalled. Some  
10 examples of services:

#### Domain-Data Manager

The Tracker invokes the Domain-Data Manager service on every browser  
page. It communicates with the server to get relevant data and scripts needed to  
fill the forms. The data is stored in the data collection object and used by other  
15 services. This service only deals with data maintenance, and hence other  
services do not have to do the same.

#### Automated Form-Filling

The Tracker invokes the form-filling service when the browser is at a form.  
It uses domain data collection to get relevant data and scripts needed to fill the  
20 forms. For specialized services like changing the credit card used in the  
transaction, the service provider can use scripts from the Script Collection object.  
(A more detailed discussion of the form-filling service is discussed below).

#### Transaction Tracking

The transaction tracking service provider recognizes and sends the  
25 transaction related pages to the server while a user is purchasing an item. The  
server extracts the transaction information, stores it for future referral and uses it  
to compute the cash back.

The foregoing services are for example purposes only and are not  
intended to limit the range of services that may be provided through the BCA.

5 Other service providers can be added, as desired. For additional example, other  
services could include a site listing service that provides a list of sites related to  
the domain where a user is browsing; a newsgroups or surveys service that  
provides newsgroups or surveys related to the content of the page being  
10 displayed; a comparison shopping or pricing service that prices an item on a web  
page or provides comparison shopping information. Comparison shopping  
information could include price of the same item at other stores or competitive  
alternatives to the particular item. The input for the comparison shopping search  
engine could be implemented by parsing a web page on a browser for keywords  
or elements that match predetermined keywords or elements in a database.  
15 More details of such a system are provided in commonly assigned US Pat. App.  
App. No. 09/881353, filed June 12, 2001, entitled "Systems And Methods For  
Automatic Hyperlinking Of Words Or Other Data Items And For Information  
Retrieval Using Hyperlinked Words Or Data Items", the entire disclosure of which  
is hereby incorporated by reference for all purposes.

### **AUTOMATED FORM FILLING SERVICES**

20 Conventional online forms are not associated with any kind of intelligent  
application for comprehensive and accurate form filling. Because the fields in  
forms presented from one Web site to another often call for the same data, users  
25 are faced with repetitive and time-consuming data input tasks. The form filling  
service of the present invention generally relates to the automated input of data  
into fields or forms or other means for receiving data that may be presented to  
users of a computer network such as the Internet. The present invention  
particularly relates to client and server applications that enable automated form  
30 filling for online shopping.

5 In one possible embodiment of automated form filling, knowledge of a  
form's structure may be developed manually or through a "network effect" of  
obtaining the necessary data for form filling. In other words, knowledge of the  
structure of a given form may be obtained by monitoring the actions of a plurality  
of users filling out the form manually. Once sufficient knowledge is obtained, the  
10 relevant data on the forms structure may be sent to the form filling module  
associated with a BCA. The network effect system accordingly is based on  
having users participate in an online program that supports automated form  
filling. When more users visit particular sites, the system will "learn" more about  
that site, such as the form names, form control data, etc., and subsequently  
15 apply this learned information in form filling for other users.

***System Introduction:***

The automated form filling process involves two parts: identifying (1) what  
to fill, and (2) where to fill a user's information. The "what to fill" for a particular  
20 user is provided by a user when he or she registers on a Web site sponsoring  
automated form filling. The kind of information that can be provided includes  
names, addresses, birth dates, social security numbers, phone numbers, fax  
numbers, email address, and any other kind of information for form fields. The  
form filling module uses this information and populate the specific site variables  
25 for actual form filling.

Since form filling on a particular site will not work if mapping information is  
not present in the system, the part of "where to fill" requires the information about  
the particular site, such as URLs, form names, variable names and the mapping  
of these variables to the data that a user has already provided. Gathering this  
30 data used to be accomplished manually by a mapper who would scan each site,  
locating each form and mapping the site's variable names with database fields

- 5 for a user's data. With the use of network effect it is now possible for the system to learn about the site as more and more people fill out forms on the site. Fig. 10 is a flowchart displaying a scheme for this network effect.

**Form filling Process:**

- 10 The control information (the form names, frame names, variable names, URLs, etc.) obtained from a site must be mapped to a user's data that the system already stores. This data is collected, for example, from members participating in an infomediary Web site program. This means all the controls of the site should be mapped to some common name with which the value in a user's table can be
- 15 dynamically associated. For example, if the site *s* has a text field *t1*, which should be filled with a first name, then there will be an entry in one of the tables for site *s* and variable *t1* mapping value of *firstname*. The actual *firstname* value is picked from a user's login table and is associated with the variable *t1* by the backend form filling data generator. The following table (Table 1) shows an example list of
- 20 such mapping identifiers used in one possible embodiment of the system of the present invention. This example is not intended to be limiting, and persons in the art will appreciate from this disclosure that other embodiments and mapping identifiers are possible.

**Table 1**

Billing mapping	Shipping mapping	Description
bill:fullname	ship:fullname	Full Name ( John Smith)
bill:first_name	ship:first_name	First Name (John)
bill:middle_name	ship:middle_name	Middle Name (H)
bill:last_name	ship:last_name	Last Name (Smith)
bill:gender	ship:gender	Male/Female
bill:suffix	ship:suffix	Sr
bill:email	ship:email	Complete email (John@eboodle.com)
bill:email_domain	Ship:email_domain	Domain part of email

		(Eboodle.com)
bill:email_userid	Ship:email_userid	John
bill:company	Ship:company	EBoodle Inc
bill:pobox	Ship:pobox	
bill:street1	ship:street1	275 Landing Dr
bill:street2	ship:street2	
bill:city	ship:city	Mountain View
bill:state:short	ship:state:short	CA
bill:state:full	ship:state:full	California
bill:state:name_of_the_vendor	ship:state:name_of_the_vendor	CA-California
bill:zipcode	ship:zipcode	94043
bill:province	ship:province	
bill:country:name_of_the_vendor	ship:country:name_of_the_vendor	USA – United States
bill:country:abbrev	ship:country:abbrev	USA
ship:country:full	ship:country:full	United States
Bill:country:short	ship:country:short	US
bill:local_phone	ship:local_phone	456-6789
bill:local_phone_plain	ship:local_phone_plain	4566789
bill:ext_phone	ship:ext_phone	88888
bill:areacode_phone	ship:areacode_phone	123
bill:left_phone	ship:left_phone	123
bill:mid_phone	ship:mid_phone	456
bill:right_phone	ship:right_phone	6789
bill:phone_dashes	ship:phone_dashes	123-456-6789
bill:phone_plain	ship:phone_plain	1234566789
bill:local_fax_dashes	ship:local_fax_dashes	456-6789
bill:local_fax_plain	ship:local_fax_plain	4566789
bill:fax_dashes	ship:fax_dashes	1234566789
bill:fax_plain	ship:fax_plain	123-456-6789
bill:left_fax	ship:left_fax	123
bill:mid_fax	ship:mid_fax	456
bill:right_fax	ship:right_fax	6789

5



- 5 Entries in columns 1 and 2 represent the keywords used to represent a user's information. These keywords are called 'db\_global"s. Because these keywords also represent the format information, it is possible that there might be multiple db\_globals for single user data. The following tables (Table 2 and 3) list db\_globals for different user data formats.

10

**Table 2**

cc:exp:%%mm/yy	09/00
cc:exp:%%mm/yyyy	092000
cc:exp:%%mmyy	0900
cc:month:%%full	September
cc:month:%%multi	Sept
cc:month:%%short	09
cc:month:%%single	9
cc:name:%%:	John Smith
cc:num:%%cc1	4244
cc:num:%%cc2	4244
cc:num:%%cc3	4244
cc:num:%%cc4	4244
cc:num:%%dashes	4244-4244- 4244-4244
cc:num:%%plain	424442444244 4244
cc:num:%%spaces	4244 4244 4244 4244
cc:year:%%yy	00
cc:year:%%yyyy	2000
cc:zip:%%:	94043

**Table 3**

login:passwd	
login:userld	
title_prefix	Mr.
Name	John Smith
Firstname	John
Midname	H
Lastname	Smith

Mother_maiden	Mary
Suffix	Sr.,Jr.
Gender	M/F
emailid	John@eboodle.com
emailid_domain	Eboodle.com
emailid_user	John
areacode_eve_phone	123
ext_eve_phone	8888
eve_phone_dashes	123-456-6789
eve_phone_plain	1234566789
left_eve_phone	123
mid_eve_phone	456
right_eve_phone	6789
local_day_phone	456-6789
local_day_phone_plain	4566789
areacode_day_phone	123
ext_day_phone	8888
day_phone_dashes	123-456-6789
day_phone_plain	1234566789
left_day_phone	123
mid_day_phone	456
right_day_phone	6789
local_eve_phone	456-6789
Local_eve_phone_plain	4566789
fax_dashes	123-456-6789
fax_plain	1234566789
left_fax	123
mid_fax	456
right_fax	6789
soc_sec_plain	339956398
soc_sec_dashes	123-45-6789
soc_sec_left	123
soc_sec_mid	45
Soc_sec_right	6789

fax_dashes
fax_plain
left_fax
mid_fax
right_fax
soc_sec_plain
soc_sec_dashes
soc_sec_left
soc_sec_mid
soc_sec_right
soc_sec_right

5

**BCA Input:**

Whenever the BCA identifies a site where form filling information is not available, it sends out the following information to the server. Each parameter corresponds to the variable name on the site.

Var0 = Form name:form num:control type:var\_name:var-value:seq-control:control # within seq:db\_global:description  
Var1 = Form name:form num:control type:var\_name:var-value:seq-control:control # within seq:db\_global:description

....  
Seq0 = total # of controls in sequence.  
Seq1 = total # of controls

....

The form name is the name of the form on the site's page. The form number is the position of the form with respect to the top of the page. Control type can be text if the variable is an input box, radio for a radio button, and list for a list box. Variable value is what a user enters in that field, sequence control represents the number of groups of variables of a specific type, and control number within a sequence represents the location of the current control in the sequence of controls. In certain situations, depending on some of the keywords

5 on a page, the BCA can identify what the variable corresponds to. In such  
situations, the db\_global field is set to one of the mapping identifiers listed in the  
above table. Based on the context, the BCA can sometimes get the description  
associated with a particular control. If the BCA gathers this information, it is sent  
to the description field. The sequence parameters indicate the number of controls  
10 in each group of controls.

**Example:** Consider a page on a vendor site that has two forms f1 and f2  
and in f1 there are two text field controls t1, t2 where a user has entered values  
v1, v2 followed by one radio button r1 with three options r11, r12, r13 and a user  
has chosen r12, one list box l1 with l11, l12, l13 as three options, a user has  
15 picked l13 from the list and in form f2 there is one radio button r1 with two options  
r21, r22 and a user has chosen r21 followed by one input text box t1 and the  
value a user entered is v1. The BCA will send

20 Var0=f1:1:text:t1v1:1:1::  
Var1=f1:1:text:t2:v2:1:2::  
Var2=f1:1:radio:r1:r12;r11;r12;r13:2:1::  
Var3=f1:1:list:l1:l13;l11;l12;l13:3:1  
Var4=f2:2:radio:r1:r21;r21;r22:1:1  
25 Var5=f2:2:text:t1:v1:2:1  
Seq0=2 seq1=1 seq2=1 seq3=1 seq4=1

For efficiency, the data is first imported into a database table. The  
analyzing and mapping of data is then processed by the Type and Learn System  
in a batch mode.

### 30 **Type and Learn System for Automated Form Filling:**

A scheme for another embodiment of automated form filling is shown in  
Fig. 11. The scheme provides general steps used in an intelligent, automated  
form filling system to implement the network effect knowledge acquisition of a

5 form's structure. The Type and Learn scheme identifies the mapping information  
of a given variable name based on the value entered by a user. The variable  
value is checked against any of the data that is present in a user's address or  
credit card table. The comparison of the variable value and the actual data in a  
10 user's table is not a character-by-character comparison. Instead, controls like  
phone numbers or fax numbers might be represented as one text box for the  
entire number or three separate text boxes for area code, middle three digits,  
final four digits. The system is intelligent enough to check all the possible cases  
to determine what exactly a given variable represents. For credit card expiration  
15 dates, for example, the site may represent the data in a variety of formats:  
(mm/yy, mm/yyyy, three letter month name and two or four digits of year, etc.) so  
the system will check all possible combinations to learn the variable type.

***Option Controls:***

If a particular control is a radio box or a pull down list of items, then the  
form filling process needs to know all the possible options in that control. For  
20 example, credit card type is usually presented as a radio box or a list control on  
most sites. The BCA should not only gather the particular value that a user  
selected but also the other option values in that control. In certain circumstances  
the BCA will make an "intelligent choice" in identifying the type of the control.  
For example, if a pull down list has 12 items and the first entry is a number 01/1  
25 or "Jan" or "January," the system will deduce it is a date control. If the BCA is  
able to deduce the type of the control then all the option values are not sent.  
However, for any other radio/list control the BCA sends all the option values. The  
Type and Learn' system then analyzes the entire information to determine  
controls.

30

5           ***Enhancements to the system:***

          The usefulness of this system depends on how much stored user data matches with the data that is entered into a site's forms. Improvements can be made to the process by applying further fuzzy logic type of comparisons so that some common differences are ignored in the comparison. For example, in a  
10   database with user profile information, the street address data may exist as '123 Main Street' while on a site users might enter '123 Main St.' The system may be set to recognize a predetermined degree of matching even though values do not match exactly.

          Secondly, the BCA could be intelligent in identifying information about the  
15   controls on a page. Because the BCA has access to all the data on a page, it should only collect relevant data and parse it back to the server. This would reduce the amount of data transferred and improve the processing efficiency of the Type and Learn system.

20   **Type and Remember System Example of Automated Form Filling**

          Type and Remember enables the BCA to process and store user information without the need to maintain user profile data on a remote server, such as the BizRate.com website or other CCS. Instead, when a user fills in a form on a site that has already been mapped by the server, values entered by a  
25   user can be stored at that time for retrieval the next time a user needs to fill in a form. Referring to Fig. 12, the server side process is described below:

1. A list of form elements and values is sent from the client to the server.

          Each entry in the list contains the variable name of the form element and the value that a user entered into that form element.

2. For each element in the list, the form variable name is looked up in the database to identify corresponding user information (i.e. phone number, credit card number, etc.).
3. Once a user information type is determined, the value that a user entered (e.g., 650-555-0819) can be entered in a user's account information database under the column corresponding to that information type (i.e. phone number).
4. After the entire form element list is processed, the changes are saved to the database. The next time the client requests user information to fill forms, the saved information will be sent. This method alleviates a user from ever having to manually enter information at the BizRate.com or another infomediary or central computer web site.

The client browser submits the data as a post to the server in the following format:

```
vname=<name>&fURL=<URL>&data=<varName:value,varName:value>&  
addrName1=<name>&addrName2=<name>&CCName=<name>&prompt  
SaveAddrCC={yes,no}
```

The server looks up each varName in the vendor\_map table, and retrieves the db\_global value for the varName. A user-entered value for that db\_global is entered in a hashtable by the db\_global. After all values have been entered, they are extracted from the hashtable one at a time for reentry into the user tables of the database. Different processing actions are taken for each entry, based on the db\_global. For instance, if the db\_global="first\_name", the value that a user entered is simply put in the database in the "first\_name" column. However, if the db\_global="left\_phone", then the values "ship:mid\_phone" and "ship:right\_phone" are also retrieved from the hashtable, and the three entries are concatenated before being entered into the database under the "phone"

5 column. All variables for a given URL are entered on the same row in the database table. If there are two address forms on the same URL, all the "ship" db\_globals will be added as one row, and all the "bill" db\_globals will be added as a second row. The actions to be taken for each db\_global are as follows:

10       fullname  
          Separate the words into and store them in "first\_name",  
          "middle\_name", and "last\_name" in the database.

          first\_name  
          Insert in database under this name.

          middle\_name  
15       Insert in database under this name.

          last\_name  
          Insert in database under this name.

          email  
          Insert in database under this name.

20       email\_domain  
          Retrieve email\_userid as well, concatenate the two and store under  
          "email" in the database.

          email\_userid  
          Retrieve email\_domain as well, concatenate the two and store  
25       under "email" in the database.

          company  
          Insert in database under this name.

          pobox  
          Insert in database under this name.

30       street1  
          Insert in database under this name.

          street2  
          Insert in database under this name.

          city  
35       Insert in database under this name.

          state:short  
          Insert in database under "state".

          state:full  
          Use the full state map to find the short version of the state name.  
40       Store this value in the database under "state".

          state:name\_of\_the\_vendor



5            Use the specific state map for this vendor to find the short version  
             of the state name. Store this value in the database under "state".  
zipcode  
             Insert in database under this name.  
province  
10           Insert in database under this name.  
country:name\_of\_the\_vendor  
             Use the specific country map for this vendor to find the short  
             version of the country name. Store this value in the database  
             under "country".  
15           country:abbrev  
             Use the abbrev country map to find the full version of this country  
             name. Store this value in the database under "country".  
country:full  
             Store this in the database under "country".  
20           country:short  
             Use the short country map to find the full version of this country  
             name. Store this value in the database under "country".  
local\_phone  
             get left\_phone and concatenate that with this value. Store result  
25           under "phone".  
local\_phone\_plain  
             get left\_phone and concatenate that with this value, and add  
             dashes. Store result under "phone".  
ext\_phone  
30           Insert in database under this name.  
left\_phone  
             Get mid\_phone and right\_phone, or local\_phone. Concatenate and  
             store under "phone".  
mid\_phone  
35           Get left\_phone and right\_phone. Concatenate and store under  
             "phone".  
right\_phone  
             Get left\_phone and mid\_phone. Concatenate and store under  
             "phone".  
40           phone\_dashes  
             Store under "phone".  
phone\_plain  
             Add dashes, then store under "phone".

5 cc:exp:%%:mm/yy  
Using the slash, parse out the month and year. Store the month  
under "cc\_exp\_mon" and the year under "cc\_exp\_year".  
cc:exp:%%:mm/yyyy  
Using the slash, parse out the month and year. Store the month  
10 under "cc\_exp\_mon" and the year under "cc\_exp\_year".  
cc:exp:%%:mmyy  
Parse out the month and year. Store the month under  
"cc\_exp\_mon" and the year under "cc\_exp\_year".  
cc:month:%%:full  
15 Store under "cc\_exp\_mon".  
cc:month:%%:multi  
Store under "cc\_exp\_mon".  
cc:month:%%:short  
Store under "cc\_exp\_mon".  
20 cc:month:%%:single  
Store under "cc\_exp\_mon".  
cc:name:%%:  
Store under "cc\_name".  
cc:num:%%:cc1  
25 Store under "cc\_num1".  
cc:num:%%:cc2  
Store under "cc\_num2".  
cc:num:%%:cc3  
Store under "cc\_num3".  
30 cc:num:%%:cc4  
Store under "cc\_num4".  
cc:num:%%:dashes  
Parse using dashes. Store 4 values under "cc\_num1" thru  
"cc\_num4".  
35 cc:num:%%:plain  
Parse every 4 digits into separate token. Store 4 values under  
"cc\_num1" thru "cc\_num4".  
cc:num:%%:spaces  
Parse using spaces. Store 4 values under "cc\_num1" thru  
40 "cc\_num4".  
cc:year:%%:yy  
Store under "cc\_exp\_year".  
cc:year:%%:yyyy  
Store under "cc\_exp\_year".

5           cc:zip: %:  
            Store under "cc\_zip".

**Type and Learn: Browser Companion Agent Interaction with Sites and Javascripts**

10

Fig. 13 is a flow chart showing how the BCA of the present invention interacts with Web sites and JavaScripts. The browser sends "BeforeNavigate complete event" on each page. The BCA then executes the JavaScript. The JavaScript checks if the user is logged in and the page from which the user is navigating is an ecommerce form by scanning for specific keywords in the body text, such as "address", "state", "city", "zip", "visa", "card", "expiration", "date". Once the form is identified as an ecommerce form, the element collection is executed and the data is sent to the server.

15

20   **Type And Remember For Internet Explorer**

**Module Description:**

The Type and Remember feature captures user information such as address or credit card data and prompts a user to save the data to the server. The stored information can then be retrieved for automatic form filling.

25

For example, a user navigates to a vendor's Web site. When the user is ready to confirm an order, the user is required to fill out billing and shipping addresses and credit card information. Normally, the BCA can fill this data into the forms automatically if the site is supported and has stored enough user data. If the BCA cannot fill in all or a portion of the form because of insufficient user data, a user will need to type in the remaining fields in the form. In this case, the BCA captures data the user typed in and prompts them to save it. If a user elects to save the data, it sends the data to the server and it is stored in the database.

30

- 5           Note: Depending on the data that a user types in, the dialog prompt can ask a user to save any combination of information. For example, the information could be billing address, shipping address and credit card.

**Design:**

**Description**

- 10           • Request a flag from the server to indicate whether to prompt to save user typed in data.
- Request the following information from the server to current page:
- o Number of billing variables in eBForm attribute.
- o Number of shipping variables in eBForm attribute.
- 15           • Check user typed in data on every BeforeNavigate2 event if a user logged in.
- Compare user data on the form with the one in **domain data object**. If either a credit card or street address1 is different, determines whether billing address, shipping address, and/or credit card information has
- 20           changed.
- Display a prompt dialog that contains checkboxes for saving billing address, shipping address, and/or credit card.
- If a user elects to save, send the following as a packet to the server:
- o Domain name, form URL
- 25           o List of var names and values
- o Billing name, shipping name, and/or credit card name.
- The server determines variables that belong to billing, shipping, or credit card fields and saves them in the proper groups with the new names.

**Data Format:**

- 30           • Receive from the server

GANZLAW, PC  
PO Box 10105  
Portland, Oregon 97296  
Phone: (503) 224-2713  
Docket No.: BIZ/01-0008  
Express Mail Label No.: EL627039647US Deposited: August 20, 2001

- 5
  - o eBPromptSaveAddrCC
  - o eb\_addr
    - eBNumBillingVars
    - eBNumShippingVars
  - Send to server
- 10
  - o Vname=<nam>&fURL=<URL>&data=<varName:value,varName:val  
ue&billName=<name&shipName=<name>&CCName=<name>&&pr  
omptSaveAddrCC={yes,no}

As should be apparent from the foregoing descriptions of embodiments of  
15 the present invention, the BCA of the present invention discussed above enables  
“plug and play” services:

- a. The BCA may run multiple services in a given priority. The priority  
is set by an individual service. The priority can be set on the server  
side and hence can be changed whenever required.
- 20 b. The services implement common interfaces. All the common  
features and data are implemented and stored at a level higher  
than the services. The data is passed to these services.
- c. The BCA may be set up to only run those services that are  
authenticated by using a security code. Authentication is done for  
25 security reasons to ensure the data passed to the service is not  
misused and the service is a genuine service implemented by a  
company and not hackers.
- d. Service components can be individually downloaded on demand.  
This way, the downloads become light and make a user experience  
30 better as to downloading all the code at once.

- 5           e. The BCA has auto an update feature, i.e. it can upgrade itself to a newer version on detection. New services may also be easily added to the BCA.
- f. With all the above features it becomes much easier to develop and put services in production.

10           The BCA of the present invention discussed above enables caching techniques of data:

- a. A user visits different domains during browsing sessions. The BCA executes the commonly defined interfaces of the services. The services may perform different operations for each page. The architecture allows the services to store data in the collection and retrieve as and when needed.
- b. The data for each site is stored in the object collection. In an event when a user goes back to the site in the same session in memory, temporarily cached data is used.
- c. The data is stored in data objects and made available to all services. The service share data and hence redundant calls to the server to get the same data are avoided.
- d. LRU (Least Recently Used) technique is applied to store data in the collection. The collection stores 'n' number of site data. The threshold 'n' comes from the server and can be changed as needed. LRU technique allows storing data for most frequently visited sites in that session. If a user visits a new site then the data that was not used for the longest time will be replaced by the new site data in the collection.

30           The BCA of the present invention discussed above enables interactivity between sites:

- 5 a. For example, the present invention's architecture facilitates  
interactivity between, HTML from Web sites, HTML associated with  
a BCA, BCA service modules, JavaScript and MS Browser object.
- 10 b. The need for all components to interact with each other and  
maintain a level security is important. The HTML in the BCA, which  
interacts with a user, needs to communicate with main browser.  
Providing a proxy object does this. The main browser object is  
made accessible to all the objects in the same process. The  
JavaScript can access the object using the proxy object. The  
JavaScript can be the one included in the BCA or any services  
15 hosted by the companion.
- c. The communication between the CCS site and BCA is made  
possible using messaging techniques via the proxy. The site  
action/changes can be communicated to the companion using the  
messaging.
- 20 d. The BCA provides a storing mechanism for all the JavaScript  
variables as name and value pairs. In this way, as a user navigates  
from one tab to the other, the state of the variables is not lost.

25 The foregoing embodiments and features are for illustrative purposes and  
are not intended to be limiting and persons skilled in the art are capable of  
appreciating other embodiments from the scope and spirit of the foregoing  
teachings.